UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/621,951	07/17/2003	Mark Allen Grubbs	AUS920030463US1	1546

77351 7590 07/13/2010

IBM CORP. (AUS)
C/O THE LAW OFFICE OF JAMES BAUDINO, PLLC
1124 N. FIELDER ROAD
#106
ARLINGTON, TX 76012

EXAMINER	
LU, CHARLES EDWARD	
ART UNIT	**PAPER NUMBER**
2161	

MAIL DATE	DELIVERY MODE
07/13/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte MARK ALLEN GRUBBS
and GERALD FRANCIS McBREARTY

Appeal 2009-002723
Application 10/621,951[1]
Technology Center 2100

Before JOHN A. JEFFERY, LEE E. BARRETT, and THU A. DANG, *Administrative Patent Judges*.

BARRETT, *Administrative Patent Judge*.

DECISION ON APPEAL[2]

This is a decision on appeal under 35 U.S.C. § 134(a) from the non-final rejection of claims 1-20. We have jurisdiction pursuant to 35 U.S.C. § 6(b).

---

[1] Filed July 17, 2003, titled "Performance-Enhancing System and Method of Accessing File System Objects."

[2] The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, or for filing a request for rehearing, as recited in 37 C.F.R. § 41.52, begins to run from the "MAIL DATE" (paper delivery mode) or the "NOTIFICATION DATE" (electronic delivery mode) shown on the PTOL-90A cover letter attached to this decision.

We affirm-in-part.

STATEMENT OF THE CASE

The invention

The invention relates to enhancing the performance of accessing file system objects. The file system objects that are frequently being accessed are determined. Each one of these file system objects has a pathname and an inode number. The inode number is used to locate the file system object on a storage system. The pathname of each file system object and its inode number are cross-referenced and cached. Having a whole pathname of a file cross-referenced with its inode number and entered into a memory allows the inode number to be obtained with one memory access instead of the many memory accesses that are usually required.

Illustrative claim

Claim 1 is reproduced below for illustration:

> 1. A method of providing a performance-enhancing way of accessing frequently-accessed file system objects comprising the steps of:
>
> determining at least one frequently-accessed file system object in a file system upon mounting the file system at a mount point on a computer system, each file system object having a pathname and an inode number, the inode number for locating the file system object on a storage system;
>
> entering the pathname of the at least one file system object into a memory system; and

cross-referencing the pathname of the at least one file system object in the memory system with its inode number thereby enabling the inode number to be obtained with one memory access.

The references

Sinha	5,437,029	July 25, 1995
Nevarez	5,499,358	Mar. 12, 1996

S.R. Kleiman, USENIX Summer 1986 Conference , *Vnodes: An Architecture for Multiple File System Types in Sun UNIX* (June 1986) ("Kleiman").

Dan Duchamp, USENIX Summer 1994 Technical Conference, *Optimistic Lookup of Whole NFS Paths in a Single Operation* (1993) ("Duchamp").

Hal Stern, *A file by any other name*, SunWorld Online, vol. 8, no. 9 (1995) ("Stern").

Appellants' Admitted Prior Art (AAPA) that "an inode is identified by a unique number called an inode number." Spec. 1, l. 25.

The rejections

Claims 1, 4, 5, 7, 8, 11, 12, 14, 15, 18, and 19 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Duchamp, Kleiman, Sinha, and AAPA.

Claims 6, 13, and 20 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Duchamp, Kleiman, Sinha, and AAPA, further in view of Stern.

Claims 2, 3, 9, 10, 16, and 17 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Duchamp, Kleiman, Sinha, and AAPA, further in view of Nevarez.

DISCUSSION

Claims 1, 4, 5, 7, 8, 11, 12, 14, 15, 18, and 19

The rejection

The Examiner finds that Duchamp teaches determining a file system object in a file system upon mounting the file system at a mount point on a computer system, each file system object having a pathname and an inode, where the inode provide a location of the file system object on a storage system. Non-Final Off. Action ("Rej.") 2. The Examiner relies on Kleiman as teaching that a "vnode" as taught by Duchamp is an "inode." *Id.* The Examiner finds that Duchamp teaches entering the path name of the file system object into a memory system and cross-referencing the path name of the file system object in the memory system with its inode thereby enabling the inode to be obtained with one memory access. *Id.* at 3. The Examiner finds that Duchamp does not expressly teach that inodes have "inode numbers," but notes that AAPA admits that "an inode is identified by a unique number called an inode number" (Spec. 1, l. 25). *Id.* The Examiner concludes that it would have been obvious to one of ordinary skill in the art to modify Duchamp so that the path in a lookup cache is cross-referenced to the inode number to successfully access a file by using an identification number of the inode. *Id.*

The Examiner finds that Duchamp does not expressly teach a "frequently accessed object," but finds that Sinha teaches allowing a user to specify a frequently accessed object (col. 7, l. 26) and concludes that it would have been obvious to one of ordinary skill in the art to further modify

Duchamp such that frequently accessed objects specified by the user are additionally determined so as to allow the user to customize file system performance as taught by Sinha (col. 8, ll. 24-30). Rej. 3.

Contentions

Appellants note that "at mount time Duchamp is only interested in knowing whether or not the file server can handle path-lookup and not whether or not there is a file system object (frequently accessed or otherwise) in the file system." Br. 5.[3] This description of Duchamp does not argue what is missing from Duchamp.

Appellants argue that "Duchamp teaches the step of cross-referencing pathnames to vnodes and not to inodes." Br. 5.

Appellants argue that Kleiman teaches that a vnode is the independent part of an inode and therefore does not read on an inode. Br. 6.

Appellants note that Sinha teaches a pathname resolution method for providing fixed speed of file accessing in a computer network wherein each user of a node of a distributed system can selectively specify one of a plurality of modes of path name resolution for use in accessing a specific file. Br. 6. However, Appellants make no specific arguments about Sinha.

Appellants argue that there is no motivation to combine the teachings of Sinha and Kleiman with Duchamp. Br. 7.

Appellants argue that the combination of references does not teach "determining at least one frequently-accessed file system object in a file

[3] We refer to the Brief filed September 13, 2007.

system upon mounting the file system at a mount point on a computer system" (emphasis added) as recited in claim 1. Br. 7; Reply Br. 3.

Issues

Does the combination of references teach or suggest the following limitations of independent claim 1: (1) "determining at least one frequently-accessed file system object in a file system"; and (2) "each file system object having a pathname and an inode number, the inode number for locating the file system object on a storage system"?

Findings of fact

Duchamp

Duchamp describes a path-to-vnode cache at the VFS (virtual file system) level. § 2.1.

Duchamp describes that a vnode is a unique ID. § 2.2.4.

Kleiman

Kleiman describes an architecture for accommodating multiple file system implementations within the Sun UNIX kernel. P. 1, Introduction.

Kleiman describes that the kernel functionality is split into file system implementation independent and file system implementation dependent parts. P. 1, Design goals.

Kleiman describes that "[t]he file system independent inode was renamed *vnode* (virtual node). All file manipulation is done with a vnode object." P. 2, Operation.

AAPA

The Specification states that "[an] inode is identified by a unique number called an inode number." Spec. 1, l. 25.

Sinha

Sinha describes that the user of a distributed system is enabled to select one of three different modes of path name resolution when accessing a file. Col. 7, ll. 20-22. "A first mode of operation (referred to in the following as Type 1 operation), to be selected when a very high speed of file access is required (for example if the file is used very frequently by that user . . . ." Col. 7, ll. 23-26. The different modes are specified by a user command. Col. 7, ll. 41-47.

When the first mode of operation is selected, the location information for the file is entered into a name cache in correspondence with the path name of the file. Col. 7, ll. 41-51. "That [location] information will basically consist of the node identifier for the node where the object is resident, and information to be used in located the object at that node." Col. 7, ll. 60-63.

Principles of law

"[T]he test [for obviousness] is what the combined teachings of the references would have suggested to those of ordinary skill in the art." *In re Keller*, 642 F.2d 413, 425 (CCPA 1981) (citations omitted).

Analysis

Claims 1, 4, 5, 7, 8, 11, 12, 14, 15, 18, and 19 are argued as a group. Therefore, we select claim 1 as a representative claim. *See* 37 C.F.R. § 41.37(c)(1)(vii)("When multiple claims subject to the same ground of rejection are argued as a group by appellant, the Board may select a single claim from the group of claims that are argued together to decide the appeal with respect to the group of claims as to the ground of rejection on the basis of the selected claim alone.").

We agree with the Examiner that the path-to-vnode cache in Duchamp at least suggests "each file system object having a pathname and an inode number, the inode number for locating the file system object on a storage system." Kleiman states that a vnode is simply a renamed file system implementation independent inode, so we agree that a vnode in Duchamp would at least suggest an inode to one skilled in the art because an inode by another name is still an inode. In addition, Kleiman teaches that the vnode is used for accommodating multiple system implementations and one of ordinary skill in the art would appreciate that a conventional inode can be used if multiple system implementations are not required. Duchamp teaches that the vnode locates the file system object and the vnode is a unique ID, so the AAPA is not needed to teach that inodes have a number for locating file system objects; in any case, that inodes have numbers is not in dispute.

The remaining issue is whether Sinha teaches or suggests "determining at least one frequently-accessed file system object in a file system" and "entering the pathname of the at least one file system object into

a memory system." Sinha describes that the user can select a mode of path name resolution and one selected mode corresponding to frequently selected files puts the path name and location of a file in a special name cache. The Examiner states that "[e]ntering frequently accessed files must include 'determining at least one frequently accessed file system object' so that proper data can be entered." Ans. 11. The Examiner is relying on an implied claim interpretation that a user may perform the "determining" step, although in the disclosed invention the step is performed by the system. We agree with the Examiner that Sinha teaches a determining step as broadly claimed. Appellants have not shown error in this reading of Sinha onto claim 1, but only argue the combination of references would not teach the step. Accordingly, we conclude that Appellants have not shown error as to this limitation of claim 1. Appellants do not argue the separate patentability of claims 8 and 15, which recite "code means" or a "processor for processing the code data" to perform the step of determining.

Conclusion

The combination of references teaches or suggests the following limitations of independent claim 1: (1) "determining at least one frequently-accessed file system object in a file system"; and (2) "each file system object having a pathname and an inode number, the inode number for locating the file system object on a storage system." The rejection of claims 1, 4, 5, 7, 8, 11, 12, 14, 15, 18, and 19 under 35 U.S.C. § 103(a) is affirmed.

Claims 2, 3, 9, 10, 16, and 17

Appellants argue that Nevarez does not teach or suggest "obtaining from an extended attribute file a list of pathnames to be entered into the memory system, the extended attribute file being associated with the mounted file system," as recited in claim 3, because "if the user supplies the pathname of the file, there is no reason for the name of the file to be obtained from an extended attribute file." Br. 9.

The Examiner states that "Nevarez was combined with the other references so that the user can obtain the pathname from Nevarez's extended attribute file and then supply the pathname to Sinha's system (which accepts pathnames)." Ans. 12.

We agree with Appellants that there does not appear to be any reason in Nevarez or Sinha for a user who is entering a pathname in Sinha to obtain a list of pathnames from an extended attribute file. The fact that Nevarez teaches an extended attribute file containing directory entries does not, by itself, provide a reason for modifying Sinha or Duchamp. Therefore, the rejection of claims 2, 3, 9, 10, 16, and 17 is reversed.

Claims 6, 13, and 20

Appellants argue that Stern does not teach or suggest "wherein the pathname of the file system object and its cross-referenced node number are removed from the memory system when the file system containing the file system object is unmounted," as recited in claim 6, because Stern teaches removing cached entries for files on a file system that is unmounted from the

DNLC (directory name lookup cache), but these entries in the DNLC are entries of edges, not pathnames as claimed. Br. 10.

The Examiner concludes that it would have been obvious to purge entries in the cache of Duchamp when the file system is unmounted "to clean up the memory system, since the cache entries would be invalid when the file system is unmounted." Rej. 5.

We agree with the Examiner that although Stern does not teach removing entire pathnames, it does teach removing cache entries when a file or directory is removed (page 4). One of ordinary skill in the art would have been motivated to remove the path-to-vnode cache entry in Duchamp's cache when the file system is unmounted because the entry no longer refers to an accessible file and because it creates room in the cache for other entries. Therefore, we affirm the rejection of claims 6, 13, and 20.

CONCLUSION

The rejections of claims 1, 4-8, 11-15, and 18-20 under 35 U.S.C. § 103(a) are affirmed.

The rejection of claims 2, 3, 9, 10, 16, and 17 under 35 U.S.C. § 103(a) is reversed.

Requests for extensions of time are governed by 37 C.F.R. § 1.136(b). *See* 37 C.F.R. § 41.50(f).

AFFIRMED-IN-PART

erc

IBM CORP. (AUS)
C/O MUNSCH HARDT KOPF & HARR, P.C.
3800 LINCOLN PLAZA
500 N. AKARD STREET
DALLAS, TX 75201-6659